Iraqi Journal of Statistical Sciences

www.stats.mosuljournals.com

Designing SDN Approach Using WebRTC for Low Bandwidth Over Data Communication

Manhal Mohamad Basher

Agricultural Technical College, Northern Technical University, Mosul, Iraq

Article information

Article history:
Received December 13, 2023
Accepted January 28, 2024
Available online December 1, 2024

Keywords:
Web Real-Time Communication (WebRTC) Software-Defined Networking (SDN)
Network throughput Bandwidth consumption

Correspondence:
Manhal.Mohamad Basher
manahlbasher@ntu.edu.iq

Abstract

The network must choose the best route from a list of options to support Quality of Service (QoS) for engaging real-time video applications like video conferencing and remote learning. Other network paths may connect the point of origin and the destination, but because of the network's intricate architecture and strong coupling, it is challenging to find a different route. Network topologies like Integrated Services (ISs) might not provide the best performance as long as they install the path determined by the routing protocol. This paper's main goal is to use Python programming language and VirtualBox Manager to develop numerous interactive video contributions while determining the quality of service. To choose the optimal route from a network-wide viewpoint, a novel work utilising Web Real-Time Interaction (WebRTC) technological advances with Software-Defined Networking (SDN). Additionally, it has illustrated the results of a setup and evaluates performance based on message complexity and network throughput metrics.

DOI 10.33899/iqjoss.2024.185238,

1. Introduction

In May 2011, Web Real-Time Communication (WebRTC) emerged as a cutting-edge technology. Google at the time announced a set of standards and JavaScript APIs for creating peer-to-peer (Browser-to-Browser) communications [1]. As a result, WebRTC is directly used to provide interactive communications over various data types, provided that it does so without the need for plugins, setting up, or payment. Additionally, WebRTC makes use of HTML5-based web browsers managed by Firefox and Chrome [2]. It has a benefit over other technologies because it enables developers to envision the browser as a full video chat terminal. On the other hand, researchers have suggested ways to address the drawbacks of conventional networks, like managing a network's programmable switches and implementing Software-Defined Networking (SDN), which normally consists of two sets of controllers [3]. As a result, SDN provides the option to abstract network architecture and infrastructure in software, which can subsequently be applied to various hardware and devices. However, SDN gives the network more programmability by enabling programming to define and alter the way it operates and is configured. Load balancing, or the process of efficiently dividing flow among network devices, is a crucial component of network design and management [4]. For the networks to manage the extra traffic and keep their high speeds, load balancing and multipathing are required. This research emphasizes the analysis of various methods for load balancing on SDNs. Network devices on the control plane of traditional network devices. The Network Configuration Protocol (NCP) and Simple Network Management Protocol (SNMP) can now be used to partially control the network [5]. This paper is structured as follows: section II contains the methodology, implementation, and evaluation. The conclusion and subsequent research III.

2. Methodology and Evaluation

One of the main problems with integrated services is that they need a lot of states and statements to keep up a single flow [6]. This limits the router's maximum size that uses the reservation control protocol. Another problem is that integrated services cannot choose another route throughout the network than the one that the routing protocol chooses [7]. The following specifications were created using a network topology that has eight nodes (switches) and two hosts (the transmitter and receiver). Because increasing fault tolerance and network throughput are the primary goals of data networks, network topology is used in the SDN OpenFlow protocol. A three-tiered switch construction with core, edge, and gathering switches is produced by this particular topology, which links and interacts with every switch. A graphical depiction of the network structure is shown in the figure.

Figure (1) uses the routing protocol of each node to show a topology consisting of 13 nodes. Furthermore, the routing protocol is unable to mount the data most efficiently from the sender to the recipient. Assuming that the path (switch1–switch2–switch4–switch6–switch8) represents the most effective means of sending video data from the message's sender (Host 1) to its recipient (Host 2), let's proceed. If the routers use Open Shortest Path First (OSPF), the shortest path—switch1–switch3–switch6–switch7–switch8—is the recommended one. This suggests that there are five hops in the video data's path. The source sends a path message to switch 1, which subsequently sends it to switch 8, setting up QoS on the path (switch 1-switch 3-switch6-switch7-switch8). When changing to version 8, the path message. The receiver modifies its soft state to correspond with the traffic specifications and responds with the reservation demand message after confirming that the traffic requirements in the PATH message are correct. Switch8 uses the limitations of the request traffic standard to send the request for reservation message to switch1 in a soft state after receiving it through the receiver. As illustrated in Figure (1), Switch1 sets its soft indicator to reservation demand activity and then delivers the reservation interest message back to the sender. Following the establishment of the reservation demand traffic standard, the sender installs the QoS path (switch 1–switch 3–switch 6–switch 7–switch 8).

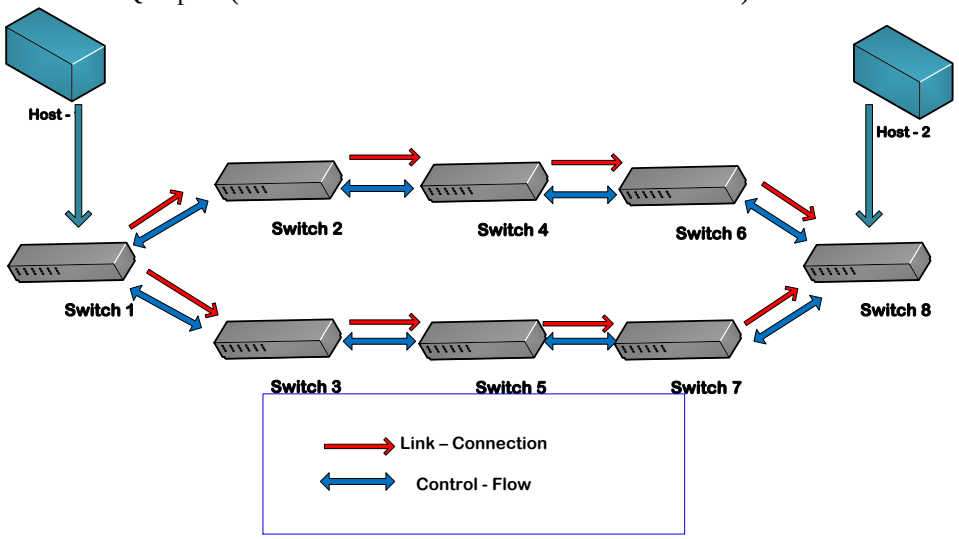

Figure1. Network Methodology

The optimal path is (switch1–switch2–switch4–switch6–switch8), as previously mentioned. Because the reserve control protocol for integrated services uses the same path as the routing protocols to set up QoS, which might have the worst performance, it is unable to select the optimal path, which is (switch1–switch2–switch4–switch6–switch8). The routing protocol finds a new shortest path if the path (switch1–switch2–switch4–switch6–switch8) fails, as illustrated in Figure (1). When the routing algorithm discovers that the link between (switch1–switch3–switch6–switch7–switch8) is broken, it will update the shortest path and update the routers' path databases to reflect the new connection. Switches 1 through 8 are the ones through which the video traffic passes. Two factors must be taken into account to select the optimal method:

1. Construct a network architecture that can decide which path is best. Using a network-wide state simplifies the process of choosing the optimal strategy. Hop-by-hop selection can lead real-time interactive video applications to choose the worst-case scenario, as shown in Figure (1). The network architecture must determine the optimal route based on the QoS requirements of the video program.

2. Develop a control system that lets interactive, real-time video apps ask the network for quality of service constantly. A protocol must be in place for real-time interactive video apps to ask the network for Quality of Service (QoS). The protocol must be straightforward and only require the states that must be tracked to guarantee that the best course is chosen.

3. Network Construction

The network concept, which gathers data about each network device, is a crucial part of the network architecture design and is necessary to address Issue 1. Referring back to the earlier example, the network-wide view design decision allowed for the resolution of Issue 1 through the use of SDN in conjunction with the OpenFlow protocol. The SDN controller is now shown in Figure (2) to more correctly represent our suggested network architecture in Figure (1). The OpenFlow protocol is used for communication between the data plane and the SDN controller also referred to as the control plane. This inquiry has made use of a ray controller. A video traffic control system with an eight-node topology runs inside the SDN controller. The SDN manager can see the entire network from the sender to the recipient. The SDN manager chooses the optimal route based on performance [9]. In addition, Figure (2) shows the architecture and the interactions between the different parts of the design (simply, only switches 1 and 2 are displayed). The sender, receiver, controller, and switch are the four most important parts. In terms of QoS, the sender and user depend on switches 1 and 2. The controller uses OpenFlow to communicate with the sender and recipient while switching over secure channels. Both the sender and the user ask the network for Quality of Service. However, switch1 and switch2 are edge switches that use packet shapers to separate the traffic coming from the sender and the traffic going to the recipient. Because of this, network traffic between switches 1 and 2 may go via several intermediate switches. Nevertheless, only the edge switches change the way network traffic flows. However, the only switches that alter the way network traffic flows are the edge switches.

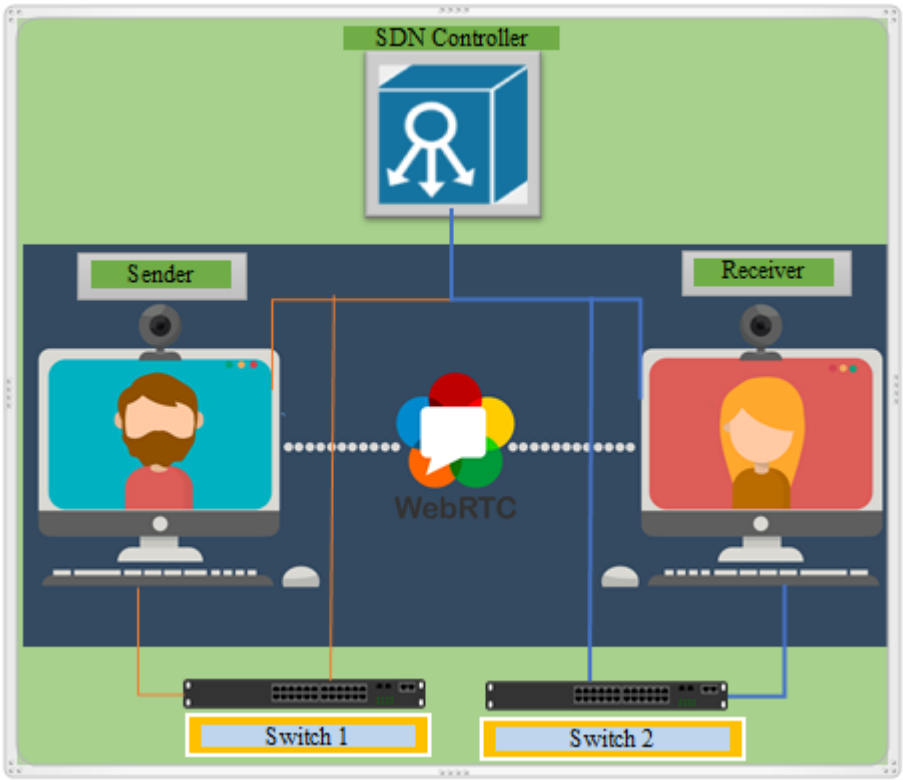

Figure2. Proposed Network Architecture

Finding each network path that operates inside the topology and figuring out which path is the best, shortest, and cheapest way to route the massive flow successfully is the goal of the routing algorithm. The switches will control each flow through

them once they are pushed. The algorithm also needs to calculate the amount of data that is sent and received on the switch ports to determine which path has the lowest total cost. Lastly, it needs to figure out how much each path will cost to compute. Ultimately, the model employs three distinct routing strategies, which it refers to as a short test path with bandwidth consumption, a quick test path with hop, and a hybrid for both. It is important to keep in mind that a request can be made by the sender or the recipient. Among the tactics included in the suggested model protocol are the following ones: The strategies are (a) controller, (b) switch, (c) receiver, and (t) transmitter.

4. Architecture Implementation

The OpenFlow switch's controller in this project, it is easy to implement the Quality of Service (QoS) function via the web page and notify the controller because the QoS module is already in place [10]. The proposed model can predefine a few QoS queues and then send requests with the appropriate data type, e.g., "the source address is 10.0.0.1." The switch would add QoS rules following user requirements when they matched. In the end, this application can continuously use bandwidth while providing users with a reliable and consistent service.

Using a mininet, the hosts, edges, and nodes from the preceding section are brought together to form the network topology [11]. The SDN controller will act as the interface controller during the network topology development process, allowing packet routing in that network design. The network topology in this project is set up based on the following requirements: two switches (c), eight hosts (b), and an SDN controller (a). Since managing problems and accelerating the network are the two main goals of data networks, the SDN Open flow protocol makes use of a network structure. In this topology, the switch architecture consists of two layers: edge switches, core, aggregation, and switches. Every switch is linked together.

5. Topology Generation

The controller in software-defined networking is responsible for storing all network topology information. The suggested technique is then used to determine which possible network topology has the shortest distance between points. The ping all command, when used with Mininet, verifies that all hosts are connected and are active and reachable from one another. To confirm that each host in the system has access to every other host, use this command. When the hosts are active, all packets are received by them, and the proportion of packets dropped is zero. To successfully connect two hosts, the ping all command may take up to an hour and a half, or even longer. In terms of data packet transmission, the bandwidth of the links connecting the hosts and switches is 0.2 Mbit, 0.1 Mbit, and 0.05 Mbit, in that order. For every routing algorithm, a fat topology is created using the subsequent steps: Open the Mininet terminal, enter the path of the algorithm, and generate a fat tree model. After that, the Mininet adds 8 hosts and 8 switches, establishes links, and turns on the controller.

6. Controller and Mininet Communication

To establish a connection between the controller and the network topology, the switches and controller communicate via the default port 8080. Each switch in the topology is given a distinct port, allowing for the separate tracking of transmitted packets. Within the scope of this project, there are eight hosts connected to twenty Openflow switches. Among these twenty switches, the first four S1001 through S1004 are considered to be the core switches. The next eight switches, from S2001 to 2008, are the aggregation switches. The network is connected to a total of eight hosts through the eight edge switches that follow. To determine the shortest path feasible within the network topology, the controller script can be run. As shown in Figure (3), all handler events and topology switches are loaded to send and receive packets between the switches.

Evaluation

The OpenFlow controller and routers communicate with each other through OSPF, which is used in SDN to determine the best path first. The packet data is stored by the controllers that use the Address Resolution Protocol. because the controller can function in two ways: proactively and reactively. A controller receives data from switches in the proactive setting while switches forward the required protocol for address resolution to the controller in the reactive mode to identify the best path. In this section evaluates the video that was submitted for the HOST001 and HOST008 projects. The "HOST001 to HOST008" and "HOST008 to HOST001" paths have the same number of hops5, which is the same. However, there is a difference in the way that packets pass through the switches. To run a quick test path with the hop in the Mininet terminal, it has done the following:

1. Enter the hop directory along with the brief test path.
2. Use a defamed WebRTC model to run a brief test path.

Architecture Model

The video spread for HOST001 and HOST008 has been evaluated in this section. The bandwidth consumption of each link between hosts HOST001 and HOST008 is known. The best path for packet transmission, however, is the one with the shortest wait time. The algorithm and the recommended model are run by the Mininet terminal to run a quick test path with the hop. The next step is to ping every network using the ICMP protocol. In Figure (3), the ICMP procedure is demonstrated.

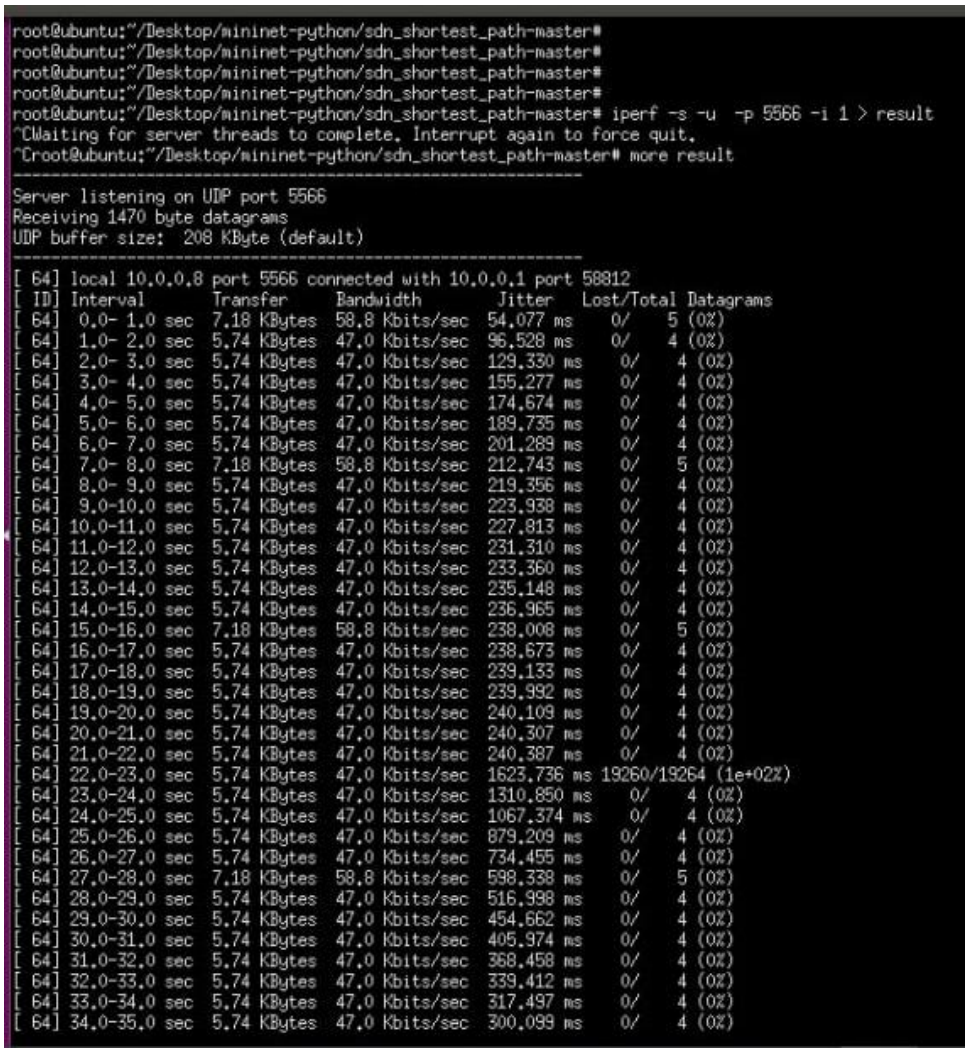

Figure3. Development of running for all network.

The controller then uses ping to find the shortest path after doing a make-discovery for each network. Controller-make-make-host1-client-generate-Tcp-traffic-delay is the algorithm that comes next. Lastly, controller-make-host1-client is the algorithm that creates the UDPP traffic bandwidth consumption.

7. Measurement of TCP and UDP Throughput

Two hosts, HOST001 and HOST008, trade packets via TCP with one another. HOST001 transmits UDP messages to HOST008 at a rate of 10M while in client mode. It calculates the amount of bandwidth and time needed for the data transfer. Additionally, the sent-message totals are computed. In server mode, HOST008 employs the same bandwidth calculations while accepting UDP packets. Consequently, we can state that the median throughput is 133 Kbits/sec, or 281

Kbytes, between 0 and 16.5 seconds. Only 298 data packets of messages were lost after ten tries. Measurements have been made of the UDP packet transmission for the WebRTC algorithm. The WRTSDN Algorithm's UDP packet transmission is displayed in Figure (4&5) which shows how WebRTC transmits UDP packets.

The amount of milliseconds that a browser must spend sending an inquiry to a server and then patiently waiting for an answer is known as the round-trip time (RTT). It is an essential indicator of performance for web applications and a key factor in determining page load time and network latency.

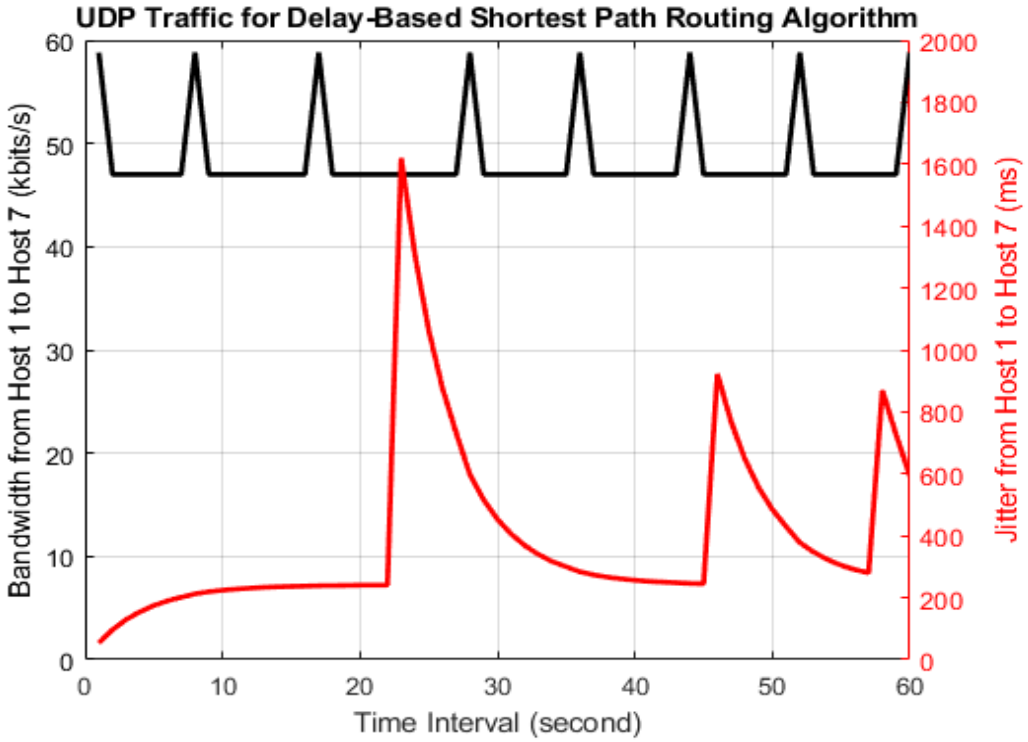

Figure5.WebRTC transmits UDP packets

8. Conclusion

As this work has shown, the primary goals of data networks are to boost network throughput and fault tolerance. In this sense, network topology is used by the SDN OpenFlow protocol. This particular topology gives rise to a three-tiered switch architecture with aggregation, edge, and core switches that are all connected to and conversing with one another. Furthermore, OSPF which interacts with every router through the OpenFlow controller—is utilized in SDN to identify the best path first. The packet data is stored by the controllers that use the Address Resolution Protocol. Switches determine the optimal path in the reactive mode and forward the required Address Resolution Protocol to the controller; switches in the proactive mode provide information to the controller. To choose the best route from a network-wide perspective, a unique study employing Web Real-Time Communication and Software-Defined Networking (SDN) technology has been finished.

9. References

1. R. Pyla, V. Pandalaneni, P. J. N. Raju, and G. P. G, “Design and Development of swarm AGV ’ s alliance for Search and Rescue operations,” *J. Robot. Control*, vol. 4, no. 6, pp. 791–807, 2023, doi: 10.18196/jrc.v4i6.18392.
2. N. Edan and S. A. Mahmood, “Design and implement a new mechanism for audio , video and screen Design and implement a new mechanism for audio , video and screen recording based on WebRTC technology,” *Int. J. Electr. Comput. Eng.*, vol. 10, no. March, pp. 2773–2778, 2020, doi: 10.11591/ijece.v10i3.pp2773-2778.
3. K. Aggarwal, “Cloud-to-Edge Internet of Things and 5G Network Security via Software-Defined Networking (2020),” *J. Crit. Rev.*, vol. 07, no. 09, pp. 3817–3825, 2023, doi: 10.48047/jcr.07.09.586.

4. B. Goswami, M. Kulkarni, and J. Paulose, "A Survey on P4 Challenges in Software Defined Networks: P4 Programming," *IEEE Access*, vol. 11, no. May, pp. 54373–54387, 2023, doi: 10.1109/ACCESS.2023.3275756.
5. Salah S. Abed, "Optimization Flow Control In Software-Defined Networking Using Cuckoo Search Algorithm," *J. Namibian Stud.*, vol. 3, pp. 2536–2553, 2023.
6. T. von Eicken, D. E. Culler, S. C. Goldstein, and K. E. Schauser, "Active messages: a mechanism for integrated communication and computation," *ACM SIGARCH Comput. Archit. News*, vol. 20, no. 2, pp. 256–266, 1992, doi: 10.1145/146628.140382.
7. R. Ramalingam *et al.*, "Routing Protocol for MANET Based on QoS-Aware Service Composition with Dynamic Secured Broker Selection," *Electron.*, vol. 11, no. 17, p. 160, 2022, doi: 10.3390/electronics11172637.
8. H. Eltaief, A. El Kamel, and H. Youssef, "MSA-SDMN: multicast source authentication scheme for multi-domain software defined mobile networks," *J. Inf. Telecommun.*, pp. 1–24, 2023, doi: 10.1080/24751839.2023.2250123.
9. Y. Al-Dunainawi, B. R. Al-Kaseem, and H. S. Al-Raweshidy, "Optimized Artificial Intelligence Model for DDoS Detection in SDN Environment," *IEEE Access*, vol. 11, no. August, pp. 106733–106748, 2023, doi: 10.1109/ACCESS.2023.3319214.
10. D. Zeman, F. Rezac, M. Voznak, and J. Rozhon, "Session Initiation Protocol Proxy in a Role of a Quality of Service Control Application in Software-Defined Networks," *Designs*, vol. 6, no. 6, pp. 1–12, 2022, doi: 10.3390/designs6060123.
11. S. Qureshi, "Path Establishment in Software Defined Optical Networks," Technology Sydney Faculty, 2023.

تصميم نهج SDN باستخدام WebRTC لنطاق ترددي منخفض عبر اتصالات البيانات

منهل محمد بشير

الكلية التقنية الزراعية، الجامعة التقنية الشمالية، الموصل، العراق.

manahlbasher@ntu.edu.iq

الخلاصة

يجب أن تختار الشبكة أفضل طريق من قائمة الخيارات لدعم جودة الخدمة (QoS) لإشراك تطبيقات الفيديو في الوقت الفعلي مثل مؤتمرات الفيديو والتعلم عن بعد. قد تربط مسارات الشبكة الأخرى نقطة الأصل والوجهة، ولكن نظرًا لبنية الشبكة المعقدة والاقتران القوي، فمن الصعب العثور على مسار مختلف. قد لا توفر طبولوجيا الشبكة مثل الخدمات المتكاملة (ISs) أفضل أداء طالما أنها تقوم بتثبيت المسار الذي يحدده بروتوكول التوجيه. الهدف الرئيسي لهذه الورقة هو استخدام لغة البرمجة Python و VirtualBox Manager لتطوير العديد من مساهمات الفيديو التفاعلية مع تحديد جودة الخدمة. لاختيار الطريق الأمثل من وجهة نظر على مستوى الشبكة، عمل جديد يستخدم التقدم التكنولوجي لتفاعل الويب في الوقت الحقيقي (WebRTC) مع الشبكات المعرفة بالبرمجيات (SDN). بالإضافة إلى ذلك، فقد أوضح نتائج الإعداد وقام بتقييم الأداء استنادًا إلى مدى تعقيد الرسالة ومقاييس إنتاجية الشبكة.

الكلمات المفتاحية: شبكة الاتصالات في الوقت الحقيقي (ويبرتك) الشبكات المعرفة بالبرمجيات (شبكة التنمية المستدامة) ، استهلاك عرض النطاق الترددي الإنتاجية للشبكة.